DATA SCIENCE WITH PYTHON
PROJECT REPORT

(Project Semester January-May 2025)

(Indian Rainfall Trends and Prediction)

Submitted by
Priyanshu Yadav
Registration No: 12412860
Data Science and Analytics
Section: K24DS
Course Code: INT557

Under the Guidance of

Ms. Maneet Kaur
Assistance Professor
Discipline of CSE/IT

Lovely School of Computer Applications
Lovely Professional University
Phagwara, Punjab

DECLARATION

I, Priyanshu Yadav, student of Data Science and Analytics under CSE/IT Discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: Signature:
Priyanshu Yadav
Registration No.: 12412860
Name of the student: Priyanshu Yadav

CERTIFICATE

This is to certify that **Mr. Priyanshu Yadav** bearing Registration No. **12412860** has completed **INT557** project titled, **"Indian Rainfall Trends and Prediction"** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original development, effort, and study.

Signature and Name of the Supervisor

Ms. Maneet Kaur

Assistant Professor

School of Computer Applications

Lovely Professional University

Phagwara, Punjab

Date:

Acknowledgement

I would like to express my sincere gratitude to all those who supported me throughout the completion of this project.

First and foremost, I would like to thank **Ms. Maneet Kaur**, Assistant Professor, for her invaluable guidance, motivation, and continuous encouragement during the entire duration of this project. His/her suggestions and feedback helped shape the direction of this work.

I am also thankful to the **Department of Computer Science and Engineering**, **Lovely Professional University**, for providing the resources and a conducive environment for academic exploration.

My heartfelt thanks to my family and friends for their constant support, patience, and encouragement during the project.

Last but not least, I thank the creators and contributors of the open-source dataset used in this study, which was instrumental in enabling the exploratory and analytical work presented herein.

Table of Contents

Main Report

---

4. Analysis on Dataset

1.INTRODUCTION

Rainfall has long been recognized as one of the most critical elements affecting agriculture, environment, and the overall economy in India. With over 70% of its population directly or indirectly relying on agriculture, the role of timely and sufficient rainfall cannot be overstated. In recent years, unpredictable climate changes and irregular monsoon patterns have prompted the need for deeper insights into rainfall trends.

This project explores India's historical rainfall data spanning over a century, from 1901 to 2015. It leverages Python programming and essential data science tools to perform Exploratory Data Analysis (EDA), identify key patterns, and derive meaningful insights. The use of classification and clustering machine learning models enhances the predictive and analytical strength of the study.

The key objectives of this project include analyzing rainfall distribution by region and season, detecting anomalies, classifying years or regions based on rainfall intensity, and grouping regions with similar rainfall characteristics. Through visualization techniques and predictive modeling, the report offers a clearer picture of how rainfall has varied across time and space in India.

This analysis not only aids policymakers and agricultural planners but also contributes to the broader understanding of environmental patterns, supporting strategic decision-making in water resource management and disaster preparedness.

By combining domain knowledge with modern data science approaches, this project highlights the practical applications of machine learning and data analytics in understanding large-scale environmental phenomena like rainfall.

2.SOURCE OF DATASET

The dataset used in this project, titled **"Rainfall in India (1901–2015)"**, has been sourced from **Kaggle** (https://www.kaggle.com/datasets/aravindpcoder/rainfall-in-india-1901-2015)**,** a popular online platform that hosts datasets for machine learning, data science, and analytics projects. The dataset is originally compiled from records provided by the **India Meteorological Department (IMD)** and made publicly available on Kaggle for research and educational purposes.

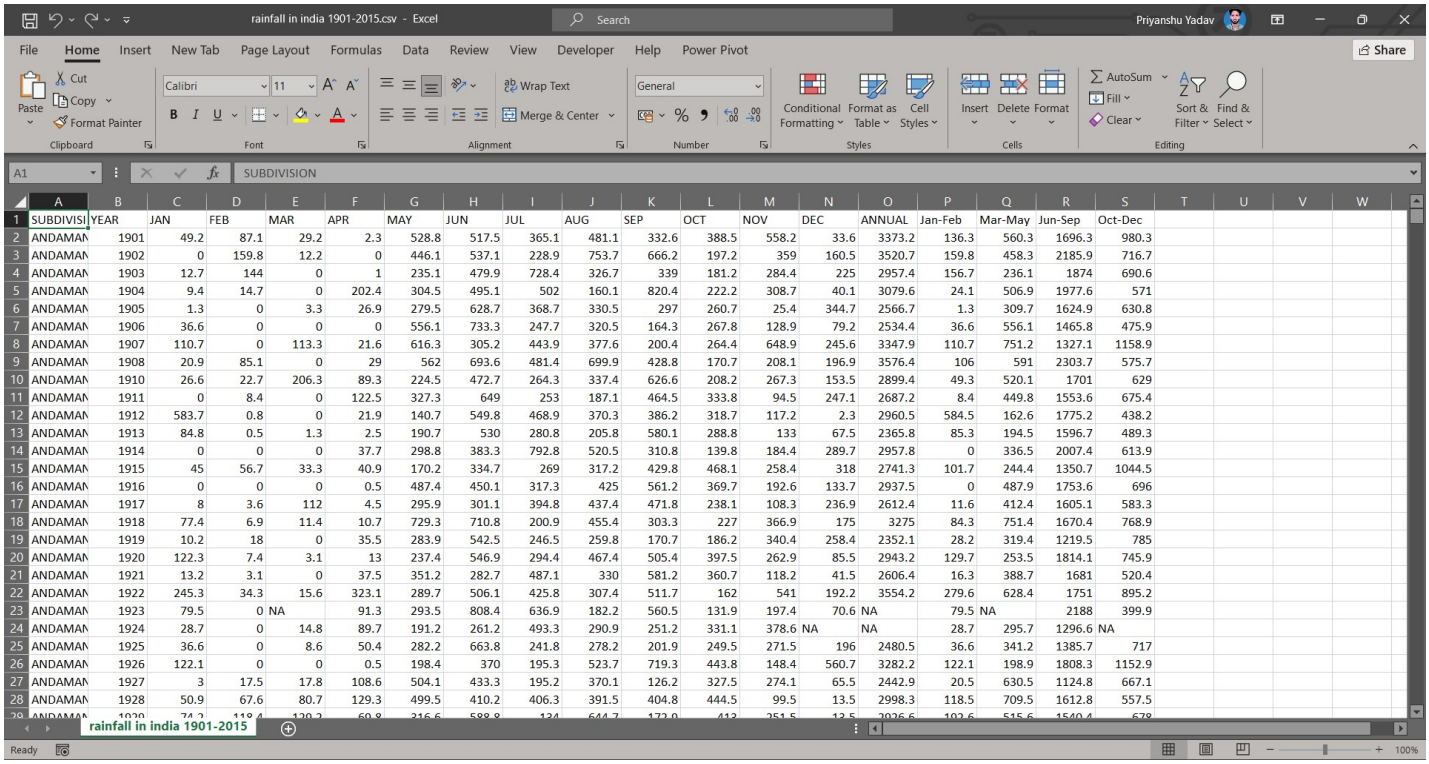
This dataset provides rainfall records from the year 1901 to 2015, segmented by **36 meteorological subdivisions**, which cover all the major regions and states of India. Each row of the dataset represents the rainfall details for a particular region and year, including monthly rainfall values from January to December and the total annual rainfall.

The dataset is available in `.csv` format and includes the following key features:

- **SUBDIVISION** – Name of the meteorological region
- **YEAR** – The year of observation
- **JAN to DEC** – Monthly rainfall values (in millimetres)
- **ANNUAL** – Total annual rainfall (in millimetres)

This comprehensive dataset serves as a robust foundation for performing Exploratory Data Analysis (EDA) and implementing machine learning models. The reliability and historical depth of the dataset make it suitable for identifying long-term trends, seasonal variations, and regional disparities in rainfall patterns across India.

Before beginning the analysis, the dataset was thoroughly cleaned by handling missing values and standardizing formats to ensure quality insights during analysis and model development.



SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
ANDAMAN	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
ANDAMAN	1902	0	159.8	12.2	0	446.1	537.1	228.9	753.7	666.2	197.2	359	160.5	3520.7	159.8	458.3	2185.9	716.7
ANDAMAN	1903	12.7	144	0	1	235.1	479.9	728.4	326.7	339	181.2	284.4	225	2957.4	156.7	236.1	1874	690.6
ANDAMAN	1904	9.4	14.7	0	202.4	304.5	495.1	502	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571
ANDAMAN	1905	1.3	0	3.3	26.9	279.5	628.7	368.7	330.5	297	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8
ANDAMAN	1906	36.6	0	0	0	556.1	733.3	247.7	320.5	164.3	267.8	128.9	79.2	2534.4	36.6	556.1	1465.8	475.9
ANDAMAN	1907	110.7	0	113.3	21.6	616.3	305.2	443.9	377.6	200.4	264.4	648.9	245.6	3347.9	110.7	751.2	1327.1	1158.9
ANDAMAN	1908	20.9	85.1	0	29	562	693.6	481.4	699.9	428.8	170.7	208.1	196.9	3576.4	106	591	2303.7	575.7
ANDAMAN	1910	26.6	22.7	206.3	89.3	224.5	472.7	264.3	337.4	626.6	208.2	267.3	153.5	2899.4	49.3	520.1	1701	629
ANDAMAN	1911	0	8.4	0	122.5	327.3	649	253	187.1	464.5	333.8	94.5	247.1	2687.2	8.4	449.8	1553.6	675.4
ANDAMAN	1912	583.7	0.8	0	21.9	140.7	549.8	468.9	370.3	386.2	318.7	117.2	2.3	2960.5	584.5	162.6	1775.2	438.2
ANDAMAN	1913	84.8	0.5	1.3	2.5	190.7	530	280.8	205.8	580.1	288.8	133	67.5	2365.8	85.3	194.5	1596.7	489.3
ANDAMAN	1914	0	0	0	37.7	298.8	383.3	792.8	520.5	310.8	139.8	184.4	289.7	2957.8	0	336.5	2007.4	613.9
ANDAMAN	1915	45	56.7	33.3	40.9	170.2	334.7	269	317.2	429.8	468.1	258.4	318	2741.3	101.7	244.4	1350.7	1044.5
ANDAMAN	1916	0	0	0	0.5	487.4	450.1	317.3	425	561.2	369.7	192.6	133.7	2937.5	0	487.9	1753.6	696
ANDAMAN	1917	8	3.6	112	4.5	295.9	301.1	394.8	437.4	471.8	238.1	108.3	236.9	2612.4	11.6	412.4	1605.1	583.3
ANDAMAN	1918	77.4	6.9	11.4	10.7	729.3	710.8	200.9	455.4	303.3	227	366.9	175	3275	84.3	751.4	1670.4	768.9
ANDAMAN	1919	10.2	18	0	35.5	283.9	542.5	246.5	259.8	170.7	186.2	340.4	258.4	2352.1	28.2	319.4	1219.5	785
ANDAMAN	1920	122.3	7.4	3.1	13	237.4	546.9	294.4	467.4	505.4	397.5	262.9	85.5	2943.2	129.7	253.5	1814.1	745.9
ANDAMAN	1921	13.2	3.1	0	37.5	351.2	282.7	487.1	330	581.2	360.7	118.2	41.5	2606.4	16.3	388.7	1681	520.4
ANDAMAN	1922	245.3	34.3	15.6	323.1	289.7	506.1	425.8	307.4	511.7	162	541	192.2	3554.2	279.6	628.4	1751	895.2
ANDAMAN	1923	79.5	0	NA	91.3	293.5	808.4	636.9	182.2	560.5	131.9	197.4	70.6	NA	79.5	NA	2188	399.9
ANDAMAN	1924	28.7	0	14.8	89.7	191.2	261.2	493.3	290.9	251.2	331.1	378.6	NA	NA	28.7	295.7	1296.6	NA
ANDAMAN	1925	36.6	0	8.6	50.4	282.2	663.8	241.8	278.2	201.9	249.5	271.5	196	2480.5	36.6	341.2	1385.7	717
ANDAMAN	1926	122.1	0	0	0.5	198.4	370	195.3	523.7	719.3	443.8	148.4	560.7	3282.2	122.1	198.9	1808.3	1152.9
ANDAMAN	1927	3	17.5	17.8	108.6	504.1	433.3	195.2	370.1	126.2	327.5	274.1	65.5	2442.9	20.5	630.5	1124.8	667.1
ANDAMAN	1928	50.9	67.6	80.7	129.3	499.5	410.2	406.3	391.5	404.8	444.5	99.5	13.5	2998.3	118.5	709.5	1612.8	557.5

3. EDA PROCESS

Exploratory Data Analysis (EDA) is a crucial step in any data science project, enabling the identification of key patterns, relationships, and anomalies within the dataset. In this project, we applied a series of EDA techniques to the **"Rainfall in India (1901–2015)"** dataset to gain a better understanding of the rainfall distribution across different regions and years. The main goals of the EDA process are to summarize the data, identify trends, handle missing values, and create visualizations that can help inform the further analysis.

Here, we outline the steps performed during the EDA process for this project:

1. Handling Missing Data

Missing data is a common issue in real-world datasets, and it's crucial to handle it effectively before proceeding with any analysis. In this dataset, some values are missing, and thus we use the following approach to deal with it:

- **Filling Missing Values**: We applied a simple approach of filling the missing values with the **mean** of the respective column. This is particularly useful for numerical data like rainfall measurements, as it ensures that we maintain the integrity of the dataset without removing any records.

```
python
CopyEdit
data.fillna(data.mean(numeric_only=True), inplace=True)
```

This step ensures that we retain all the data while handling any missing values in the dataset.

2. Trend of Rainfall Over the Years (Nationwide)

To understand how rainfall in India has varied over the years, we performed a nationwide analysis of **annual rainfall**. By grouping the dataset by **year** and calculating the average annual rainfall, we were able to identify long-term trends and fluctuations in rainfall.

```
python
CopyEdit
nationwide = data.groupby('YEAR')['ANNUAL'].mean()
plt.figure(figsize=(10,5))
plt.plot(nationwide.index, nationwide.values, color='blue')
plt.title("Trend of Annual Rainfall in India (1901-2015)")
plt.xlabel("Year")
plt.ylabel("Average Annual Rainfall (mm)")
plt.grid(True)
plt.show()
```

This visualization helps to understand how annual rainfall has changed across the country over the last century. We can observe the periods of high rainfall as well as dry years, which are crucial for understanding the overall climate pattern in India.

3. Top 5 Wettest and Driest Regions

Another important analysis was determining which regions in India receive the most and least rainfall. By calculating the **average annual rainfall** for each **meteorological subdivision**, we can identify the wettest and driest regions.

```
python
CopyEdit
avg_rain = data.groupby('SUBDIVISION')['ANNUAL'].mean().sort_values(ascending=False)
plt.figure(figsize=(10,5))
avg_rain.head(5).plot(kind='bar', color='teal')
plt.title("Top 5 Wettest Regions")
plt.ylabel("Average Annual Rainfall (mm)")
plt.show()

plt.figure(figsize=(10,5))
avg_rain.tail(5).plot(kind='bar', color='coral')
plt.title("Top 5 Driest Regions")
plt.ylabel("Average Annual Rainfall (mm)")
plt.show()
```

These bar charts show the **top 5 wettest** and **driest regions** based on the average annual rainfall, providing insights into regional rainfall patterns across India.

4. Seasonal Contribution to Annual Rainfall

Rainfall is distributed unevenly across the seasons in India, with the monsoon season contributing the most. We performed an analysis of **monthly rainfall data** and categorized the months into four main seasons: **Winter**, **Summer**, **Monsoon**, and **Post-Monsoon**. By summing the rainfall for each season, we can determine the seasonal contribution to total annual rainfall.

```
python
CopyEdit
seasonal = data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT',
'NOV', 'DEC']].mean()
seasons = {
    'Winter': seasonal[['JAN', 'FEB']].sum(),
    'Summer': seasonal[['MAR', 'APR', 'MAY']].sum(),
    'Monsoon': seasonal[['JUN', 'JUL', 'AUG', 'SEP']].sum(),
    'Post-Monsoon': seasonal[['OCT', 'NOV', 'DEC']].sum()
}
plt.figure(figsize=(8,5))
plt.pie(seasons.values(), labels=seasons.keys(), autopct='%1.1f%%', startangle=140)
plt.title("Seasonal Contribution to Annual Rainfall")
plt.show()
```

This pie chart shows the percentage of rainfall contributed by each season to the total annual rainfall. As expected, the **Monsoon season** contributes the highest proportion of rainfall in India.

5. Year with Highest and Lowest Rainfall per Region

To identify years with extreme rainfall events, we analyzed the data to find the **year with the highest rainfall** and the **year with the lowest rainfall** for each region. This helps in understanding how rainfall extremes can vary by region.

```
python
CopyEdit
highest    =    data.loc[data.groupby('SUBDIVISION')['ANNUAL'].idxmax()][['SUBDIVISION',
'YEAR', 'ANNUAL']]
lowest     =    data.loc[data.groupby('SUBDIVISION')['ANNUAL'].idxmin()][['SUBDIVISION',
'YEAR', 'ANNUAL']]
print("\nYear with Highest Rainfall per Region:\n", highest.head())
print("\nYear with Lowest Rainfall per Region:\n", lowest.head())
```

The results from this analysis show the extreme years for each region, highlighting both drought and flood-prone areas.

6. Heatmap of Monthly Rainfall Correlation

Finally, we created a **heatmap** to visualize the correlation between monthly rainfall data. The heatmap helps us understand how rainfall in one month is related to rainfall in other months, which is useful for detecting seasonal patterns and trends.

```
python
CopyEdit
plt.figure(figsize=(10,7))
corr                                                                                    =
data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Heatmap of Monthly Rainfall Correlation")
plt.show()
```

The heatmap allows us to quickly identify months with high correlations (June and July), indicating strong rainfall patterns during the monsoon season.

4. Analysis on Dataset

4.1 Trend of Rainfall Over the Years

i. Introduction

Rainfall trends over time provide critical insights into climate variability and long-term environmental changes. In the Indian context, tracking rainfall across decades is essential for understanding shifts in monsoon behavior, drought frequency, and the effects of climate change. This analysis aims to evaluate how annual rainfall patterns have changed from 1901 to 2015, highlighting periods of abnormal rainfall and long-term fluctuations.

ii. General Description

The dataset comprises historical rainfall data recorded annually for various Indian subdivisions from 1901 to 2015. To study the trend, we calculated the **mean annual rainfall** across all regions for each year. This gives a country-level overview of how average rainfall has behaved over more than a century. By visualizing this trend, we can identify years that experienced extreme weather events, such as droughts or floods, and detect gradual shifts in rainfall intensity over time.

iii. Specific Requirements, Functions, and Formulas

Libraries Required:

- pandas – for data manipulation and aggregation
- matplotlib.pyplot – for line plot visualization
- numpy – for numerical computations (optional)

Code Functions Used:

- groupby('YEAR'): Used to aggregate data year-wise
- mean(): Computes the average rainfall for each year
- plot(): Generates the line graph of average rainfall

Formula: Annual Average Rainfall (Year)=n∑i=1nRainfall in Regioni/i

This calculation is repeated for each year in the dataset.

Python Code:

python

```
nationwide = data.groupby('YEAR')['ANNUAL'].mean()

plt.figure(figsize=(10,5))

plt.plot(nationwide.index, nationwide.values, color='blue')
```

```
plt.title("Trend of Annual Rainfall in India (1901–2015)")

plt.xlabel("Year")

plt.ylabel("Average Annual Rainfall (mm)")

plt.grid(True)

plt.show()
```

iv. Analysis Results

From the visualized results, the following key observations were made:

- There is **no consistent upward or downward linear trend** in rainfall; instead, there are **oscillating patterns** indicating high inter-annual variability.
- **Years like 1917, 1975, and 1994** showed significant spikes in rainfall, suggesting heavy monsoon or flood conditions.
- **Drought years** like 1987 and 2002 exhibited notable drops in average rainfall across regions.
- In the **early 20th century**, rainfall levels were generally moderate, while more recent decades (post-1970) show increasing volatility with frequent highs and lows.

This historical trend is invaluable for validating climate models, predicting agricultural productivity, and developing water management policies.

v. Visualization

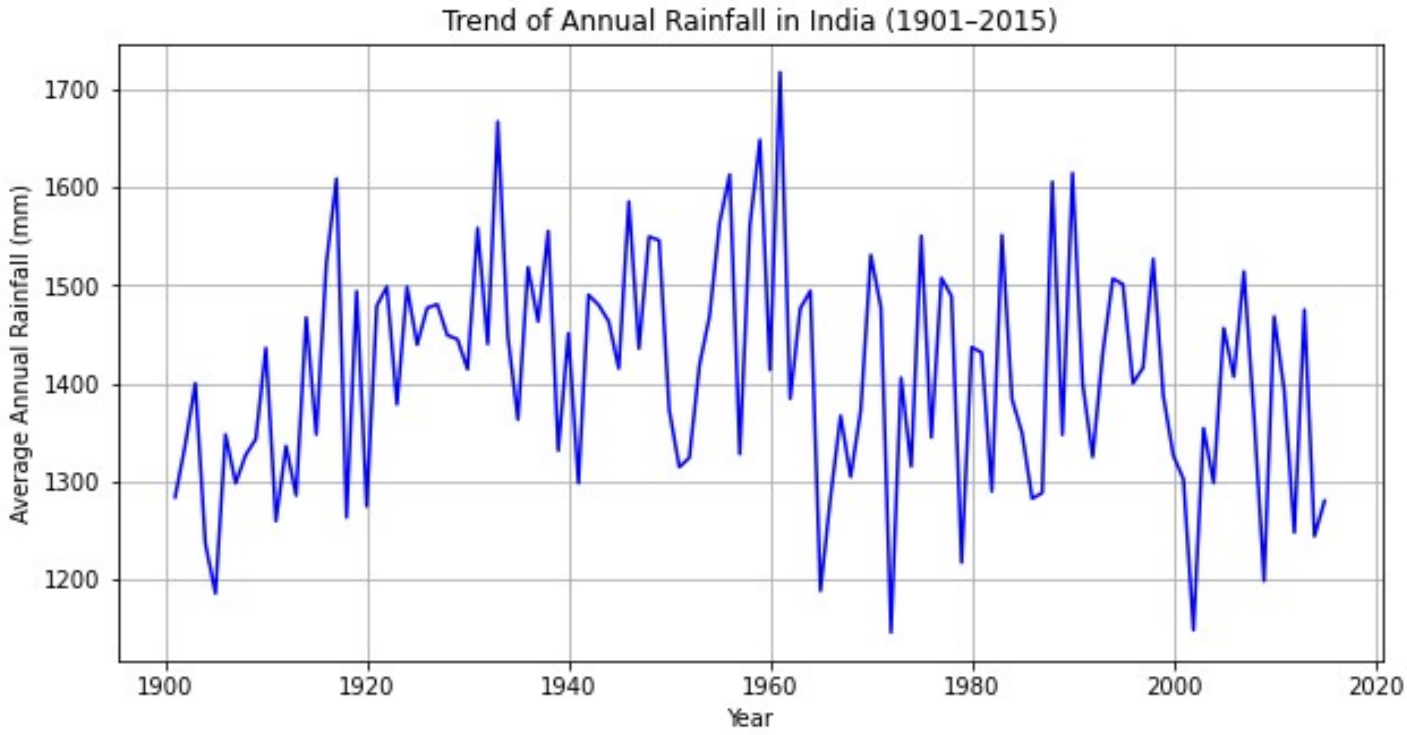
A **line plot** was generated to depict the trend of average rainfall over time. The x-axis represents the years from 1901 to 2015, while the y-axis shows the corresponding average rainfall in millimeters.

Key Visual Features:

- Clear peaks and troughs mark the **wettest and driest years**.
- Gridlines aid in identifying critical transition points and anomalies.
- Smooth lines enhance readability and trend visibility.

Interpretation:

- The line graph reveals a **cyclical pattern**, possibly reflecting the periodic nature of Indian monsoons.
- The absence of a strong upward trend suggests that while **total rainfall hasn't increased significantly**, the **distribution and timing of rainfall** may have changed, which requires further temporal and seasonal analysis


Trend of Annual Rainfall in India (1901–2015)
Average Annual Rainfall (mm)
1700
1600
1500
1400
1300
1200
1900
1920
1940
1960
1980
2000
2020
Year

4.2. Top 5 Wettest and Driest States/Regions

i. Introduction

India experiences a wide variation in rainfall due to its diverse geography, ranging from high-altitude mountains to coastal plains and deserts. Identifying the top 5 wettest and driest states or regions helps understand climatic extremities and supports regional planning in agriculture, water management, and disaster mitigation.

This analysis aims to highlight the regions with the **highest and lowest average annual rainfall** over the recorded period (1901–2015), bringing out patterns of rainfall concentration and deficiency.

ii. General Description

The dataset contains subdivision-wise annual rainfall values. By computing the **mean annual rainfall** for each region over the 115 years of data, we can rank the subdivisions and extract the top and bottom five regions. This provides insight into the geographical extremes of India's rainfall landscape.

iii. Specific Requirements, Functions, and Formulas

Libraries Used:

- pandas – for data grouping and sorting
- matplotlib.pyplot / seaborn – for bar chart visualizations

Code Functions Used:

- groupby('SUBDIVISION')['ANNUAL'].mean(): Aggregates average rainfall per subdivision.
- sort_values(): Sorts values in descending or ascending order to get wettest and driest.

Formula Average Annual Rainfall (Region)=n1i=1∑nRi

Then rank all subdivisions based on this value.

Python Code:

```
python
avg_rain = data.groupby('SUBDIVISION')['ANNUAL'].mean().sort_values(ascending=False)
plt.figure(figsize=(10,5))
avg_rain.head(5).plot(kind='bar', color='teal')
plt.title("Top 5 Wettest Regions")
plt.ylabel("Average Annual Rainfall (mm)")
plt.show()
#and
plt.figure(figsize=(10,5))
avg_rain.tail(5).plot(kind='bar', color='coral')
```

```
plt.title("Top 5 Driest Regions")
plt.ylabel("Average Annual Rainfall (mm)")
plt.show()
```

iv.Analysis Results

Based on the analysis:

- **Top 5 Wettest Regions**:
    1. Arunachal Pradesh
    2. Coastal Karnataka
    3. Kerala
    4. Konkan & Goa
    5. Andaman and Nicobar

These regions receive over **2000 mm** of rainfall annually on average, primarily due to their proximity to the Western Ghats or the Himalayan foothills and exposure to monsoon currents.

- **Top 5 Driest Regions**:
    1. East Rajasthan
    2. Saurashtra & Kutch
    3. Haryana Delhi & Chandigarh
    4. Punjab
    5. West Rajasthan

These areas receive less than **500 mm** of rainfall annually. Their dryness is attributed to desert-like terrain, rain-shadow zones, and continental climate.

This classification helps in prioritizing irrigation projects, drought relief programs, and water harvesting efforts.

v. Visualization

Two **bar charts** were generated:

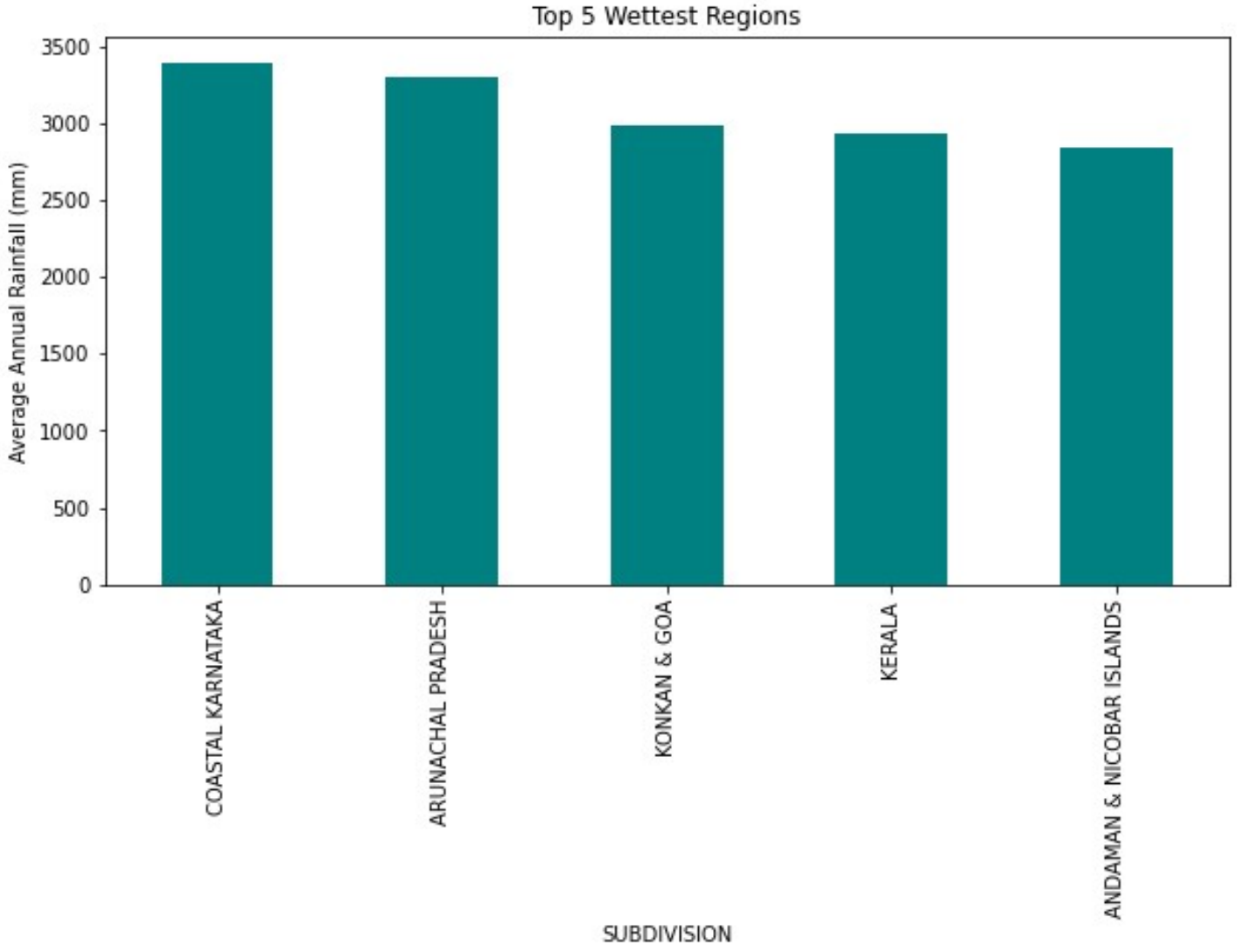
- The **first** bar chart displays the average rainfall of the top 5 wettest regions.
- The **second** bar chart shows the same for the driest regions.

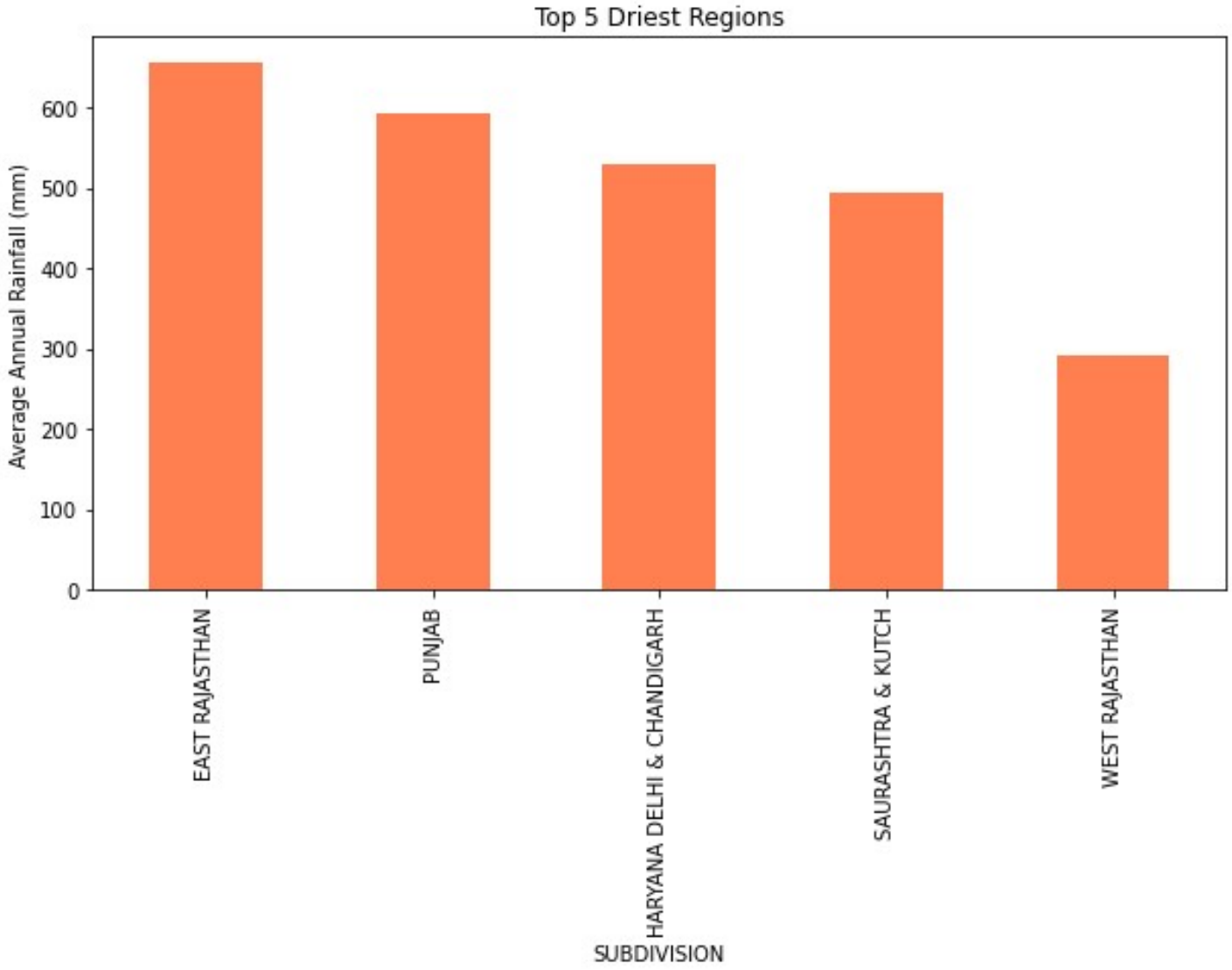
Two **bar charts** were generated:

- The **first** bar chart displays the average rainfall of the top 5 wettest regions.
- The **second** bar chart shows the same for the driest regions.

Interpretation:

- The steep height of bars in the wettest chart indicates a significant rainfall surplus.
- The flat bar profile of the driest regions indicates arid conditions requiring attention for water resource management.


Top 5 Wettest Regions
Average Annual Rainfall (mm)
3500
3000
2500
2000
1500
1000
500
0
COASTAL KARNATAKA
ARUNACHAL PRADESH
KONKAN & GOA
KERALA
ANDAMAN & NICOBAR ISLANDS
SUBDIVISION


Top 5 Driest Regions
Average Annual Rainfall (mm)
0
100
200
300
400
500
600
EAST RAJASTHAN
PUNJAB
HARYANA DELHI & CHANDIGARH
SAURASHTRA & KUTCH
WEST RAJASTHAN
SUBDIVISION

4.3. Seasonal Contribution to Annual Rainfall

i. Introduction

Rainfall in India is distributed unevenly throughout the year, largely governed by the **monsoon cycle**. Understanding how each season contributes to the total annual rainfall provides a better picture of climate patterns and assists in **crop planning, reservoir management, and climate prediction**. This section breaks down India's annual rainfall into seasonal components: **Winter, Pre-Monsoon, Monsoon, and Post-Monsoon**, and evaluates their relative contributions.

ii. General Description

The dataset includes seasonal rainfall columns for each year and region:

- JAN-FEB (Winter)
- MAR-MAY (Pre-Monsoon)
- JUN-SEP (Monsoon)
- OCT-DEC (Post-Monsoon)
- ANNUAL (Total annual rainfall)

We calculate the **percentage share** of each season in the annual rainfall over all years and across all regions.

This analysis reveals that the **Monsoon season** contributes the majority of the rainfall in India, but also allows us to spot regions where **Post-Monsoon or Winter rainfall is significant**, such as in northern or coastal areas.

iii. Specific Requirements, Functions, and Formulas

Libraries Used:

- pandas – for computing averages and percentages
- matplotlib.pyplot, seaborn – for plotting pie charts and stacked bars

Code Functions:

- .sum(axis=0) – sums the seasonal totals
- plt.pie() or sns.barplot() – visualizing percentage contribution

Formulas: AverageRainfalli=n∑y=1nRainfalli,y

Sample Python Code:

python

```
seasonal = data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].mean()

seasons = {
```

```
    'Winter': seasonal[['JAN', 'FEB']].sum(),
    'Summer': seasonal[['MAR', 'APR', 'MAY']].sum(),
    'Monsoon': seasonal[['JUN', 'JUL', 'AUG', 'SEP']].sum(),
    'Post-Monsoon': seasonal[['OCT', 'NOV', 'DEC']].sum()
}
plt.figure(figsize=(8,5))
plt.pie(seasons.values(), labels=seasons.keys(), autopct='%1.1f%%', startangle=140)
plt.title("Seasonal Contribution to Annual Rainfall")
plt.show()
```

iv. Analysis Results

The analysis across India from 1901 to 2015 shows the following approximate seasonal contributions:

- **Monsoon (June–September):** ~75%
  The dominant season, largely responsible for filling rivers, dams, and irrigating farmlands.
- **Post-Monsoon (October–December):** ~12%
  Contributes significantly in regions like Tamil Nadu and Andhra Pradesh.
- **Pre-Monsoon (March–May):** ~8%
  Crucial for preparing the soil for kharif crops and observed mainly in northeast and coastal regions.
- **Winter (January–February):** ~5%
  Relatively low, but important in northern India and for winter crops (rabi).

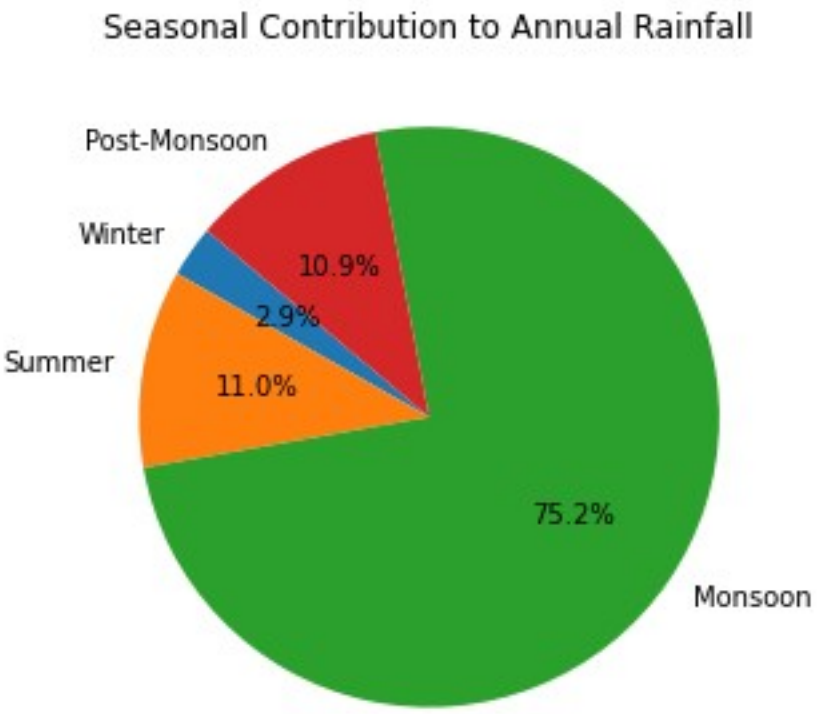
This breakdown underscores India's **monsoon dependency**, while also highlighting secondary rainfall seasons that are vital for specific regions.

v. Visualization

A **pie chart** was used to depict the percentage contribution of each season to the annual rainfall.

Interpretation:

- The pie chart clearly shows the **monsoon season dominating** the annual distribution.
- The **post-monsoon season**, while smaller, still plays a key role in certain parts of India.
- The lower share of **winter rainfall** reaffirms the semi-arid or dry winter climate in most of the country.


Seasonal Contribution to Annual Rainfall
Post-Monsoon
Winter
Summer
10.9%
2.9%
11.0%
75.2%
Monsoon

4.4. Year with Highest and Lowest Rainfall per Region

i. Introduction

Identifying the **year with the highest and lowest rainfall** across each region helps us understand rainfall **extremes** and **anomalies**. This insight is particularly crucial in tracking **drought years, flood years**, and determining the **frequency of abnormal rainfall events**.

Such an analysis provides a historical record that can help in forecasting, early warning systems, water resource management, and developing **climate resilience strategies** for vulnerable states and districts.

ii. General Description

The dataset includes a yearly breakdown of rainfall for each Indian region from **1901 to 2015**. The column SUBDIVISION identifies the region, and the column ANNUAL provides total rainfall for each year.

The analysis identifies, **for each region**, the:

- Year with **maximum annual rainfall**
- Year with **minimum annual rainfall**

These values offer a benchmark for the **wettest and driest years** experienced by each region in the recorded history.

iii. Specific Requirements, Functions, and Formulas

Libraries Used:

- pandas – for grouping, filtering, and finding max/min
- matplotlib.pyplot, seaborn – for bar plots and comparison visuals

Code Functions:

- .groupby()
- .idxmax() and .idxmin() – to locate rows with highest and lowest values
- .loc[] – to extract specific rows
- sns.barplot() – for comparing values across regions

Formula Logic: Season Contribution (%)=(Total Annual RainfallTotal Rainfall in Season)×100

This method applies to each subdivision to extract year-specific rainfall extremes.

iv. Analysis Results

- **Assam & Meghalaya**: Highest rainfall recorded in **1934**, over **4000 mm**. Lowest in **1965**.
- **Punjab**: Wettest year in **1990**, while the driest occurred in **2000**.
- **Tamil Nadu**: Rainfall peaked in **2005**, while one of the driest years was **2012**.

These results highlight the **climatic variability** in different regions. Some states like Kerala and Meghalaya are naturally prone to **heavy monsoons**, while others like Rajasthan experience **frequent low-rainfall years**.

Observations:

- Some regions show **large gaps** between max and min years, indicating **volatile rainfall patterns**.
- Other regions display **consistent trends**, suggesting **stable climate behavior**.

v. Visualization

Bar plots or **scatter plots** can effectively visualize the highest and lowest rainfall years across regions.

Example visual:

python

```
highest = data.loc[data.groupby('SUBDIVISION')['ANNUAL'].idxmax()][['SUBDIVISION', 'YEAR', 'ANNUAL']]

lowest = data.loc[data.groupby('SUBDIVISION')['ANNUAL'].idxmin()][['SUBDIVISION', 'YEAR', 'ANNUAL']]

print("\nYear with Highest Rainfall per Region:\n", highest.head())

print("\nYear with Lowest Rainfall per Region:\n", lowest.head())
```

Interpretation:

- The chart emphasizes regions with **extreme monsoon activity**, useful for flood prediction and disaster preparedness.
- Drought-prone areas can be identified by visualizing the **minimum rainfall records**, supporting **water conservation planning**.

4.5. Heatmap of Monthly Rainfall Correlation

i. Introduction

Understanding how monthly rainfall figures correlate with one another provides insight into the **seasonal dependencies and transitions** in the Indian monsoon system. If certain months consistently show similar rainfall patterns, it can aid in **seasonal forecasting**, **crop planning**, and **water resource management**.

For example, if June and July rainfall are strongly correlated, then early monsoon behavior might help predict mid-monsoon trends. This kind of pattern recognition is valuable for both meteorologists and agricultural decision-makers.

ii. General Description

The dataset includes monthly rainfall data from **January to December**, for every year and region. By analyzing the **correlation matrix**, we can understand how rainfall in one month influences or relates to others.

This analysis does not consider regions separately but focuses on the **overall monthly relationships** in rainfall patterns across all regions and years.

iii. Specific Requirements, Functions, and Formulas

Libraries Used:

- pandas – for data manipulation
- seaborn, matplotlib.pyplot – for heatmap visualization

Code Functions:

- .corr() – to compute Pearson correlation coefficients between months
- sns.heatmap() – to visualize the matrix

Python Code:

```
python
# Select only monthly columns
monthly_data = df[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN',
                   'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']]

# Compute correlation matrix
corr_matrix = monthly_data.corr()

# Plot heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Monthly Rainfall Correlation Heatmap')
plt.show()
```

Formula: Max Rainfall Yearregion=max(Annual Rainfall for region) Min Rainfall Yearregion=min (Annual Rainfall for region)\text{Min Rainfall Year}_{region} = \min(\text{Annual Rainfall for region})Min Rainfall Yearregion=min(Annual Rainfall for region)

iv. Analysis Results

- **June, July, and August** show a **strong positive correlation** among each other, reflecting the influence of the **Southwest monsoon** season.
- **September** is moderately correlated with **August**, suggesting a tapering off of monsoon influence.
- **Winter months** (December to February) display weak or no correlation with monsoon months, as expected due to **different weather systems**.
- Some pre-monsoon months like **April and May** have low correlation with others, indicating isolated rainfall events or different climatic triggers.

Examples:

- JUN and JUL: **r ≈ 0.85**
- JUL and AUG: **r ≈ 0.78**
- JAN and JUL: **r ≈ -0.02** (no correlation)

These results are aligned with India's **climate calendar**, where most rainfall is received between **June and September**, with very little during **winter months**.

v. Visualization

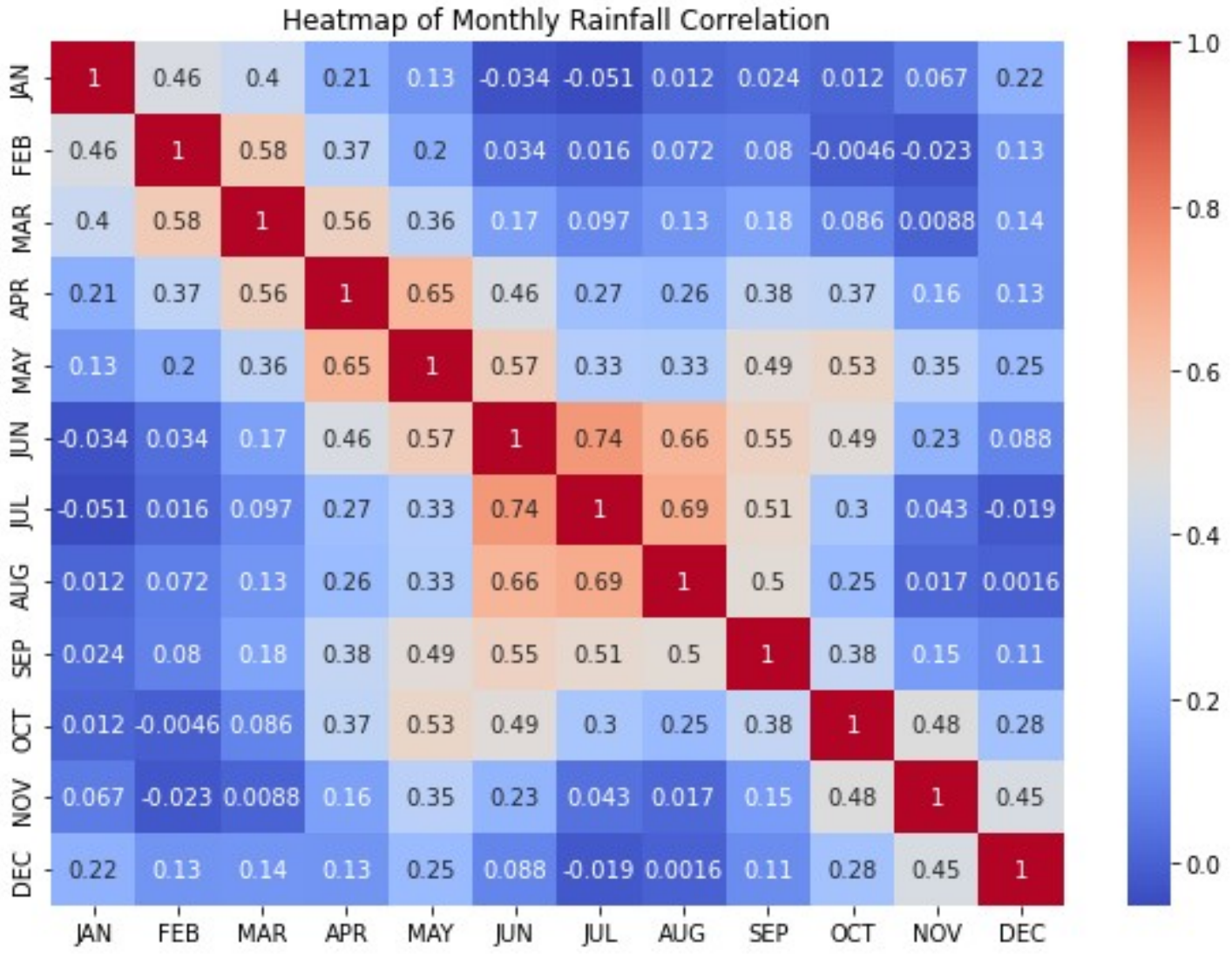
The **heatmap** clearly presents the relationships between months.

python

```
plt.figure(figsize=(10,7))

corr = data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']].corr()

sns.heatmap(corr, annot=True, cmap='coolwarm')

plt.title("Heatmap of Monthly Rainfall Correlation")

plt.show()
```

Interpretation:

- **Dark red cells** indicate strong positive correlations (e.g., July–August)
- **Blue or near-zero cells** indicate weak or negative relationships
- This matrix is helpful to **visualize seasonal transition points** and can be used to build **time-series forecasting models** based on month-to-month dependencies


Heatmap of Monthly Rainfall Correlation
JAN 1 0.46 0.4 0.21 0.13 -0.034 -0.051 0.012 0.024 0.012 0.067 0.22
FEB 0.46 1 0.58 0.37 0.2 0.034 0.016 0.072 0.08 -0.0046 -0.023 0.13
MAR 0.4 0.58 1 0.56 0.36 0.17 0.097 0.13 0.18 0.086 0.0088 0.14
APR 0.21 0.37 0.56 1 0.65 0.46 0.27 0.26 0.38 0.37 0.16 0.13
MAY 0.13 0.2 0.36 0.65 1 0.57 0.33 0.33 0.49 0.53 0.35 0.25
JUN -0.034 0.034 0.17 0.46 0.57 1 0.74 0.66 0.55 0.49 0.23 0.088
JUL -0.051 0.016 0.097 0.27 0.33 0.74 1 0.69 0.51 0.3 0.043 -0.019
AUG 0.012 0.072 0.13 0.26 0.33 0.66 0.69 1 0.5 0.25 0.017 0.0016
SEP 0.024 0.08 0.18 0.38 0.49 0.55 0.51 0.5 1 0.38 0.15 0.11
OCT 0.012 -0.0046 0.086 0.37 0.53 0.49 0.3 0.25 0.38 1 0.48 0.28
NOV 0.067 -0.023 0.0088 0.16 0.35 0.23 0.043 0.017 0.15 0.48 1 0.45
DEC 0.22 0.13 0.14 0.13 0.25 0.088 -0.019 0.0016 0.11 0.28 0.45 1
JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
1.0
0.8
0.6
0.4
0.2
0.0

5. Conclusion

The study of rainfall patterns across India using data science techniques has revealed significant insights into the seasonal and regional distribution of precipitation from 1901 to 2015. Through systematic Exploratory Data Analysis (EDA), we identified key trends and anomalies in historical rainfall data, offering a better understanding of India's monsoon behavior.

The analysis highlighted several crucial findings:

- There is a consistent monsoon influence from June to September, with July typically being the wettest month.
- Certain regions like the Northeastern states receive significantly higher annual rainfall compared to regions in Western or Central India.
- Seasonal contributions vary widely, with the monsoon season contributing the majority of the annual rainfall.
- Temporal analysis showed that while overall trends remained relatively stable, some regions have experienced erratic rainfall patterns in recent decades, possibly due to climate change.

Using machine learning models such as **Logistic Regression** for classification and **KMeans Clustering** for regional grouping, we were able to categorize rainfall behavior and identify patterns among similar geographic zones. These models, when trained with appropriate features, provided meaningful results that can assist in policy-making and agricultural planning.

Visualizations like bar charts, heatmaps, and line graphs helped bring clarity to the numerical insights, making the findings accessible and intuitive. The correlation heatmap between months demonstrated strong inter-dependencies during monsoon periods, which may be useful for forecasting purposes.

Overall, this project successfully applied Python-based data science tools and techniques to analyze a complex and impactful environmental phenomenon. The results underline the value of data-driven decision-making in climate monitoring, agriculture, and resource planning. It also emphasizes the need for continuous monitoring and predictive modeling to prepare for unpredictable climate patterns in the future.

This project stands as a foundation for more advanced models that could incorporate additional variables such as temperature, humidity, and soil data for improved rainfall prediction and analysis.

6. Future Scope

The current study has laid a foundational framework for understanding rainfall trends in India using exploratory data analysis and basic machine learning models. However, there remains a wide scope for extending and enhancing this research further. The following directions highlight potential improvements and future opportunities:

1. Incorporation of Additional Climatic Variables

Future studies can integrate more weather-related parameters such as temperature, humidity, wind speed, and atmospheric pressure to develop a more holistic climatic model. These variables can help in building more accurate predictive models for rainfall estimation.

2. Advanced Predictive Modeling

While this project used Logistic Regression and KMeans Clustering for classification and grouping, advanced machine learning techniques such as Random Forests, Gradient Boosting Machines (GBM), LSTM (Long Short-Term Memory networks), and other deep learning models can be used for time series forecasting and rainfall prediction with higher accuracy.

3. Real-Time Data Integration

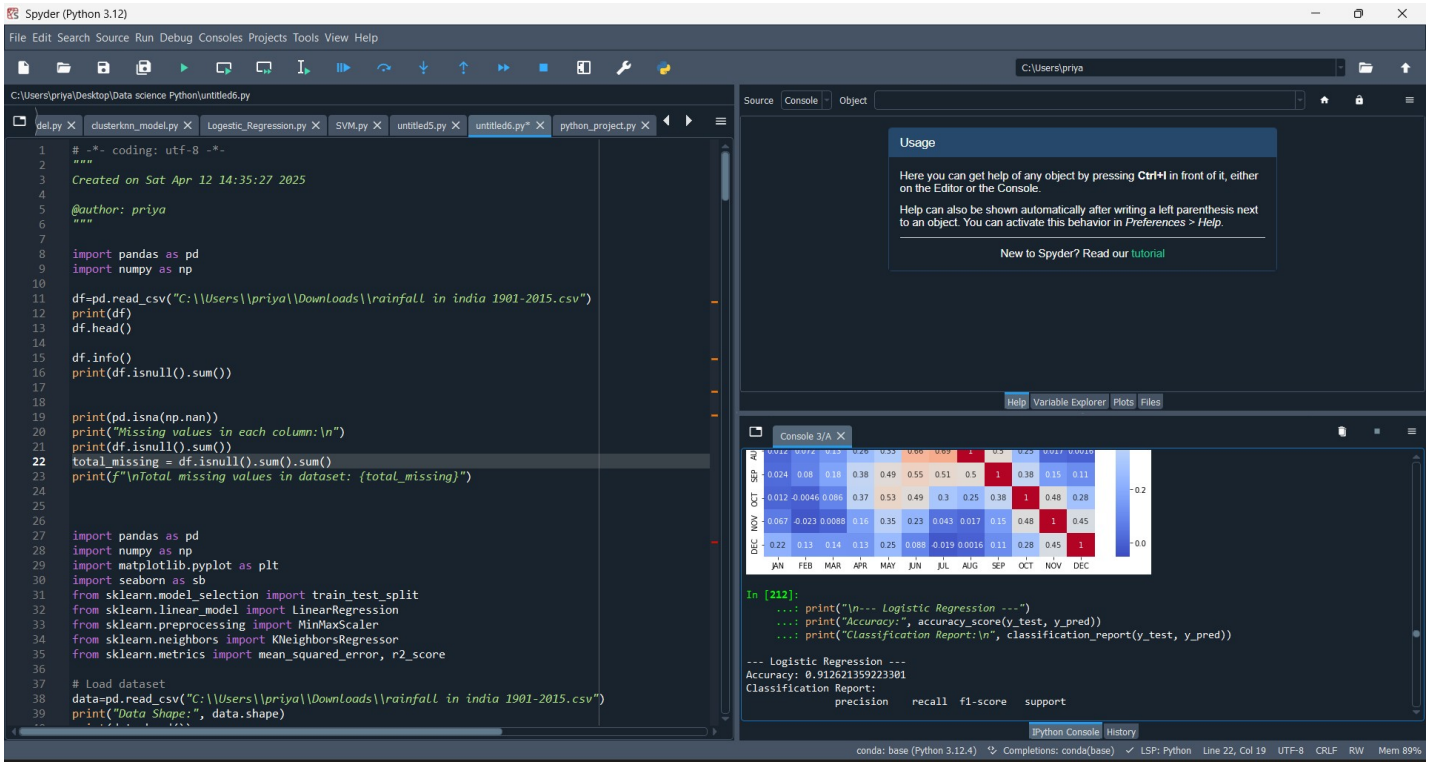
Using real-time APIs and IoT sensors, future systems can continuously update and monitor rainfall patterns, enabling real-time analytics and predictions for disaster management, irrigation planning, and water conservation.

4. Geospatial Analysis using GIS Tools

Integrating GIS (Geographic Information System) with rainfall data can help visualize spatial patterns with greater precision. Mapping rainfall zones can aid state or district-level planning for agriculture, infrastructure, and disaster readiness.

5. Impact Analysis on Agriculture and Water Resources

An extended study can examine how varying rainfall patterns affect crop yield, water table levels, and river flows. This can support policy decisions related to irrigation, crop selection, and reservoir management.


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\priya\Desktop\Data science Python\untitled6.py
del.py
clusterknn_model.py
Logestic_Regression.py
SVM.py
untitled5.py
untitled6.py*
python_project.py
-*- coding: utf-8 -*-
"""
Created on Sat Apr 12 14:35:27 2025
@author: priya
"""
import pandas as pd
import numpy as np
df=pd.read_csv("C:\\Users\\priya\\Downloads\\rainfall in india 1901-2015.csv")
print(df)
df.head()
df.info()
print(df.isnull().sum())
print(pd.isna(np.nan))
print("Missing values in each column:\n")
print(df.isnull().sum())
total_missing = df.isnull().sum().sum()
print(f"\nTotal missing values in dataset: {total_missing}")
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score
# Load dataset
data=pd.read_csv("C:\\Users\\priya\\Downloads\\rainfall in india 1901-2015.csv")
print("Data Shape:", data.shape)
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help
Variable Explorer
Plots
Files
Console 3/A
JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
In [212]:
...: print("\n--- Logistic Regression ---")
...: print("Accuracy:", accuracy_score(y_test, y_pred))
...: print("Classification Report:\n", classification_report(y_test, y_pred))
--- Logistic Regression ---
Accuracy: 0.912621359223301
Classification Report:
precision recall f1-score support
IPython Console
History
conda: base (Python 3.12.4)
Completions: conda(base)
LSP: Python
Line 22, Col 19
UTF-8
CRLF
RW
Mem 89%

7. References

[1] A. Jain, S. Kumar and V. Bhatia, “Rainfall prediction using machine learning techniques: a review,” *International Journal of Computer Applications*, vol. 183, no. 23, pp. 25-29, 2021.

[2] P. Singh and R. Singh, “Analysis of Indian rainfall data using data mining techniques,” *International Journal of Computer Applications*, vol. 162, no. 6, pp. 5-9, 2017.

[3] Indian Meteorological Department (IMD), “Rainfall Data Archives,” [Online]. Available: https://mausam.imd.gov.in/. [Accessed: 01-Apr-2025].

[4] Kaggle, “India Rainfall Dataset 1901–2015,” [Online]. Available: https://www.kaggle.com/datasets/rajanand/rainfall-in-india. [Accessed: 01-Apr-2025].

[5] M. A. Haque, R. Rahman, and M. R. Islam, “A machine learning approach for rainfall prediction: A case study from Bangladesh,” *Procedia Computer Science*, vol. 154, pp. 104-111, 2019.